In [33]:

```
#importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:

```
df =pd.read_csv(r"C:\Users\chait\Downloads\archive.zip")
df
```

Out[8]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michae 674\nLau
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johı Suite Kath
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	912 Stravenue\nD
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barne
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Rayn
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Will AP 3
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 8489\nAPO
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tra Suite 076\nJ
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallac
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Geo Apt. 509\r

5000 rows × 7 columns

In [9]:

```
df.head()
```

Out[9]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	A
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Fe 674\nLaurab
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnsor Suite 079 Kathlee
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 El Stravenue\nDani WI 0
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nF
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymon AE

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             5000 non-null   float64
 1   Avg. Area House Age          5000 non-null   float64
 2   Avg. Area Number of Rooms    5000 non-null   float64
 3   Avg. Area Number of Bedrooms 5000 non-null   float64
 4   Area Population              5000 non-null   float64
 5   Price                        5000 non-null   float64
 6   Address                      5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

In [11]:

```
df.describe()
```

Out[11]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

In [12]:

```
df.columns
```

Out[12]:

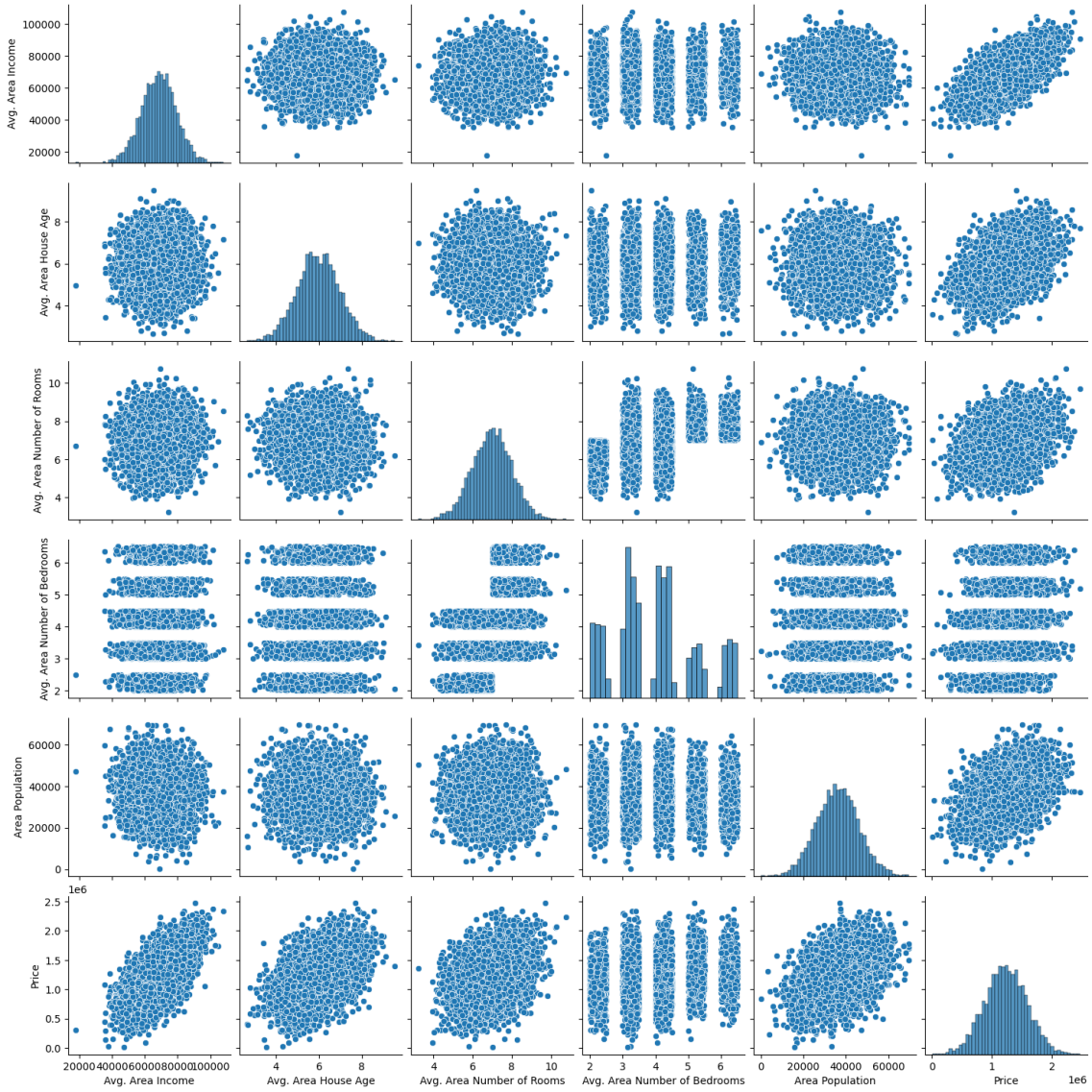
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Ro
oms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addre
ss'],
      dtype='object')
```

In [13]:

```
sns.pairplot(df)
```

Out[13]:

```
<seaborn.axisgrid.PairGrid at 0x1aa372dfd00>
```

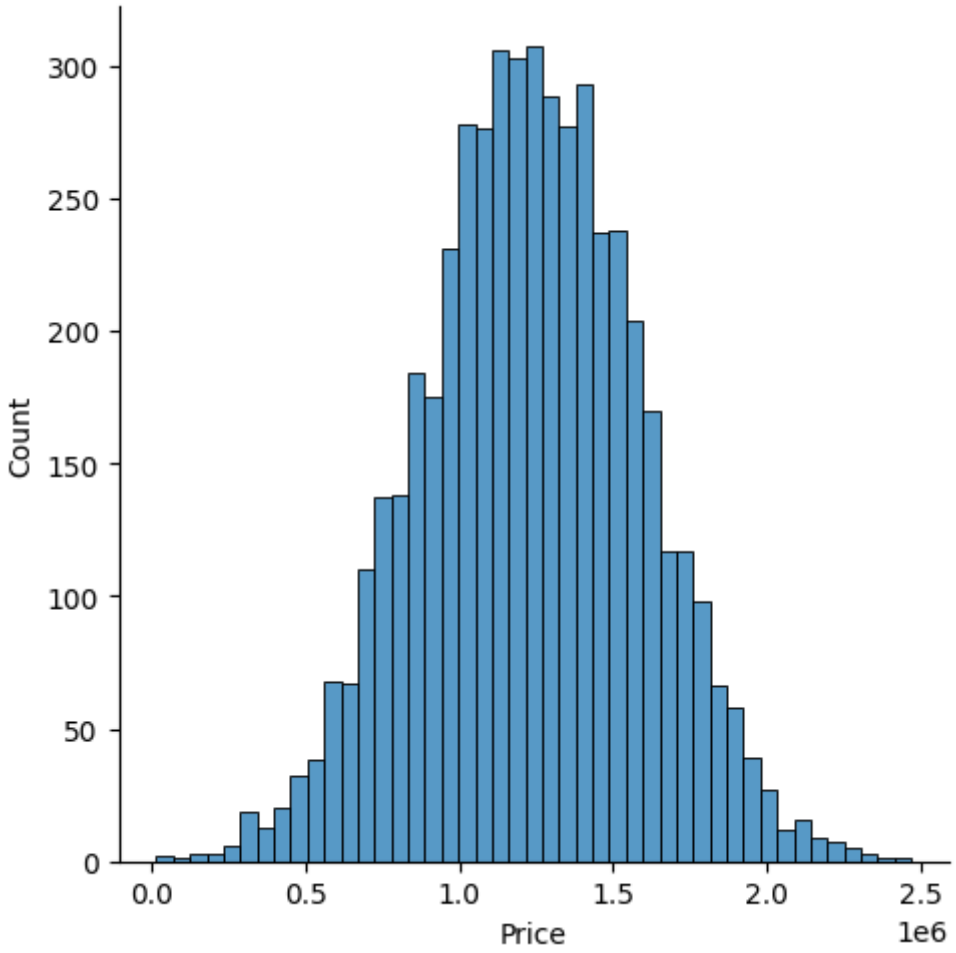


In [14]:

```
sns.displot(df['Price'])
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x1aa6bbe7970>
```

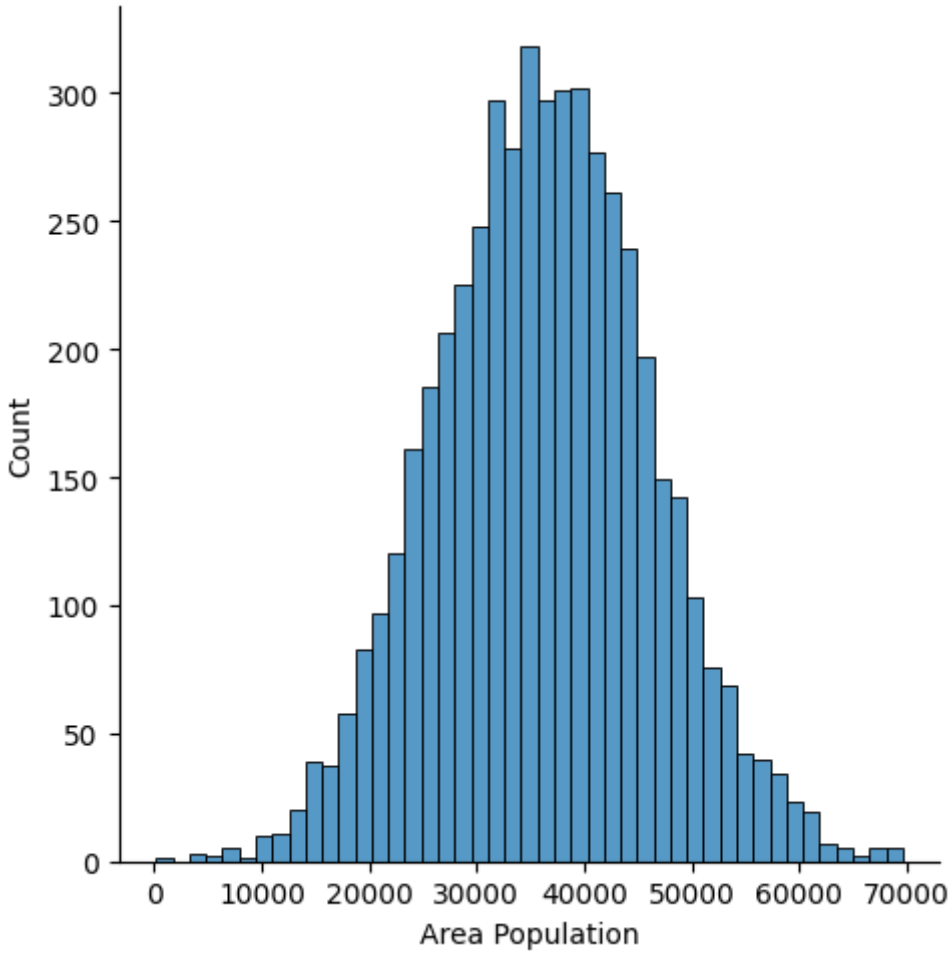


In [15]:

```
sns.displot(df['Area Population'])
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x1aa66f3c6a0>
```



In [16]:

```
Housedf= df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price']]
```

In [17]:

```
sns.heatmap(Housedf.corr())
```

Out[17]:

<Axes: >

In [22]:

```
x=Housedf[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
y=df['Price']
```

In [23]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=101)
```

In [25]:

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(x_train,y_train)
```

Out[25]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [26]:

```
print(lm.intercept_)
```

```
-2641372.6673014304
```

In [30]:

```
coeff_df=pd.DataFrame(lm.coef_,x.columns,columns=['coefficient'])
coeff_df
```

Out[30]:

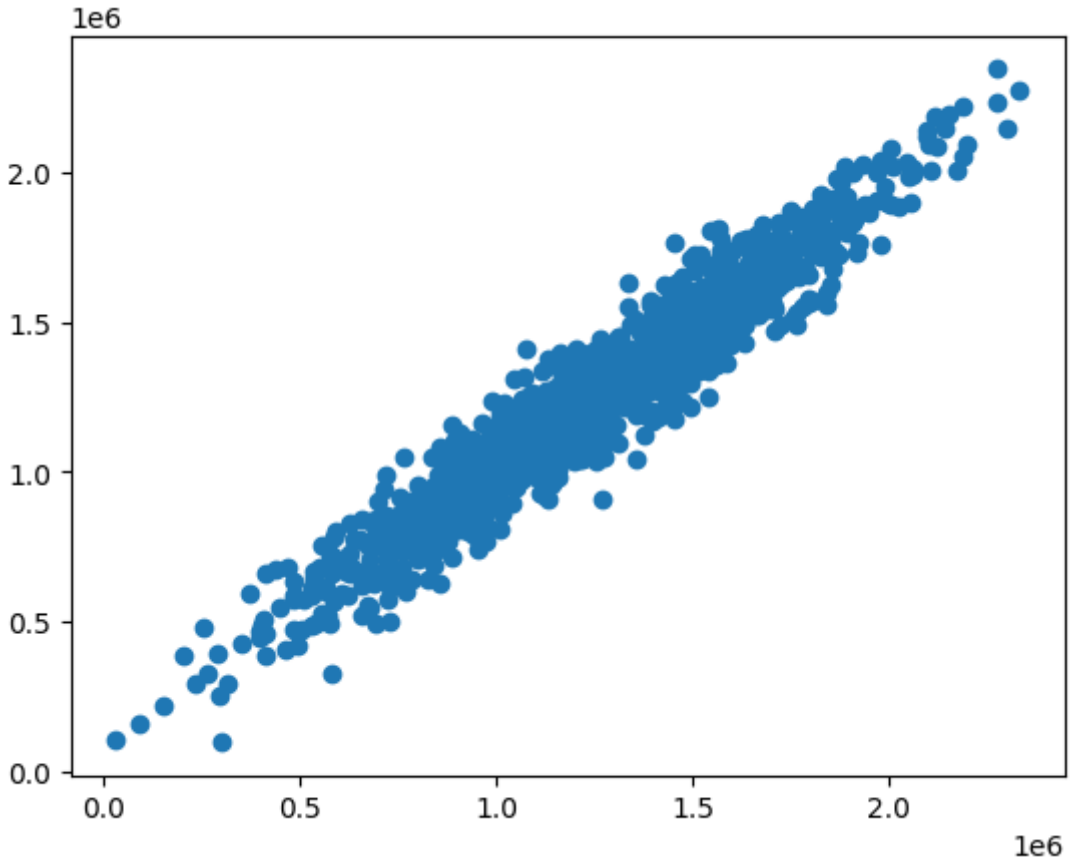
	coefficient
Avg. Area Income	21.617635
Avg. Area House Age	165221.119872
Avg. Area Number of Rooms	121405.376596
Avg. Area Number of Bedrooms	1318.718783
Area Population	15.225196

In [34]:

```
predictions =lm.predict(x_test)
plt.scatter(y_test,predictions)
```

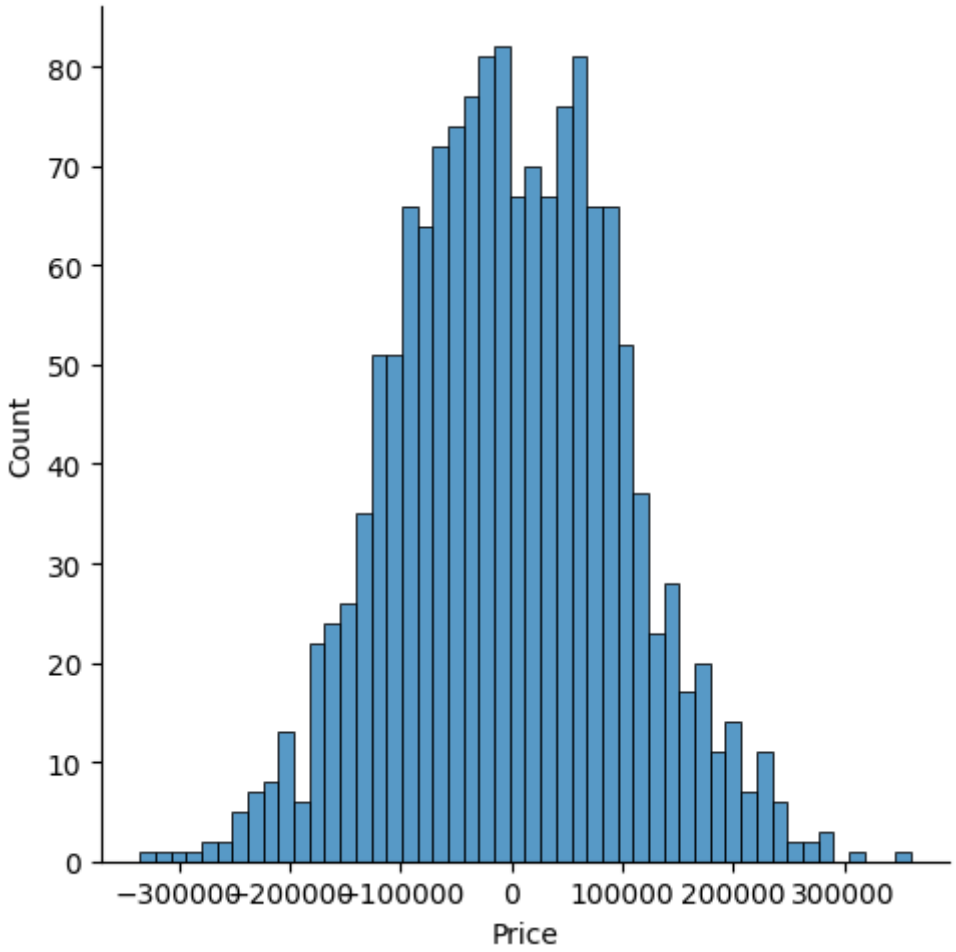
Out[34]:

```
<matplotlib.collections.PathCollection at 0x1aa70563d30>
```



In [35]:

```
sns.displot((y_test-predictions),bins=50);
```



In [36]:

```
from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(y_test,predictions))
print('MSE:',metrics.mean_squared_error(y_test,predictions))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
MAE: 81257.55795855941
MSE: 10169125565.897606
RMSE: 100842.08231635048
```

In [ ]:

```

```